IN THE CLAIMS

Please amend the claims as follows:

Claims 1-3 (Canceled).

Claim 4 (Currently Amended): A method of determining a user identification number of a colluder by detecting an embedded code which is embedded in a predetermined content and ~~concatenates~~ is a concatenation of a plurality of component codes, comprising the steps of:

receiving said predetermined content which contains the embedded code therein from an external device;

detecting said embedded code contained within said predetermined content;

dividing the embedded code into the plurality of component codes;

decoding each of the component codes, thereby to obtain a plurality of residues pairs each comprising two residues, taking a plurality of integers which are predetermined and are relatively prime to each other, as moduli;

calculating a user identification number of a colluder who made a collusion attack on the content, from the plurality of residue pairs; and

outputting said calculated user identification number, wherein

the plurality of component codes are component codes that have a possibility to have a method of decoding at least one of the residues with respect to the user identification number of the colluder.

Claim 5 (Currently Amended): A unit for determining a user identification number of a colluder by detecting an embedded code which is embedded in a predetermined content and ~~concatenates~~ is a concatenation of a plurality of component codes, comprising:

an input port configured to receive said predetermined content which contains the embedded code therein from an external device;

a detector configured to detect said embedded code contained within said predetermined content;

a code divider configured to divide the embedded code into the plurality of component codes;

a component code decoder configured to decode each of the component codes, thereby to obtain a plurality of residues pairs each comprising two residues, taking a plurality of integers which are predetermined and are relatively prime to each other, as moduli;

a colluder number calculator configured to calculate a user identification number of a colluder who made a collusion attack on the content, from the plurality of residue pairs; and

an output port configured to output said calculated user identification number, wherein

the plurality of component codes are component codes that have a possibility to have a method of decoding at least one of the residues with respect to the user identification number of the colluder.

Claim 6 (Original): A unit according to claim 5, wherein the plurality of component codes are each constructed by continuous sequences of 1 and 0, taking a predetermined number of bits as a unit.

Claim 7 (Previously Presented): A unit according to claim 5, further comprising a collusion determiner configured to determine presence or absence of a collusion from the plurality of residue pairs, wherein the colluder identification number after calculates the user identification number of the colluder, if presence of a collusion is determined by the colluder determiner.

Claim 8 (Previously Presented): A unit according to claim 5, wherein the colluder number calculator includes:

a residue selecting section for selecting one residue from each of k' inputted residues pairs, thereby to generate a set of k' residues (R1, R2, ..., Rk');

a Chinese remainder theorem section for calculating a candidate of a user identification number u of a colluder, from k residues (S1, S2, ..., Sk) which are different from each other and selected from the set of k' residues generated by the residue selecting section, in accordance with a Chinese remainder theorem; and

a consistency checking section for selecting the k residues from the set of k' residues generated by the residue selecting section, for supplying the k residues to the Chinese remainder theorem section, for specifying a user identification number of the colluder from the candidate of the user identification number u of the colluder calculated by the Chinese remainder theorem section, and for outputting the user identification number of the colluder, wherein

the consistency checking section has selection processing for selecting the k residues from the set of k' residues generated by the residue selecting section, determination processing for determining whether or not a relationship of Ri=u mod pi $(i=i_1, i_2, ..., i_\ell)$ exists between the candidate of the user identification number u of the colluder calculated by the Chinese remainder theorem section and a predetermined number (ℓ) of residues among remaining (k'-k) residues, and output processing for outputting the candidate as a user identification number of a colluder if the relationship exists as a result of the determination processing, if the relationship does not exist, a new combination of k residues (S 1, S2, ..., Sk) is selected from the set of the k' residues generated by the residue selecting section, thereby to carry out the determination processing, and if the relationship does not exist with respect to any of all combinations of k residues (S1, S2, ..., Sk), a new set of k' residues is requested to the residue selecting section, and the selection processing and the determination processing are repeated until the relationship exists.

Claims 9-13 (Canceled).

Claim 14 (Currently Amended): A unit for determining a user identification number of a colluder by detecting an embedded code, comprising:

an input port configured to receive from an external device predetermined content having the embedded code embedded therein;

a detector configured to detect said embedded code contained within said predetermined content;

a code extractor configured to extract the embedded code from the predetermined content, the embedded code ~~concatenating~~ is a concatenation of component codes respectively generated in correspondence with an inputted user identification number and also being such that among k' component codes capable of expressing all sets of integral elements that are calculated with respect to a predetermined number of user identification numbers, k combinations of the k' component codes can uniquely express the user identification numbers;

a code divider configured to perform a division and extract component codes;

a component code decoder configured to decode each of the extracted component codes;

a colluder number calculator configured to calculate a user identification number of a colluder from a decoding result of each of the component codes; and

an output port configured to output said user identification number, wherein

k' is determined to be c(k+ℓ)/q or more where c is a positive integer of 3 or more, ℓ is a positive integer, and q is a number of the integral elements which can be detected from each of the component codes when detecting the embedded code.

Claim 15 (Previously Presented): A unit according to claim 14, wherein, where pi (i=1, 2, ..., k') is a number of values which each of the integral factors calculated by the calculator can take with respect to the predetermined number of user identification numbers and where ε is a detection error rate which is assumed when detecting the embedded code, k' is determined such that a condition of

$$\left[1-\prod_{i=l}^{l}\left\{1-\left(1-\frac{1}{pi}\right)^{c}\right\}\right]^{c(k+l)/2^{C_{k+l}}X\,2^{k+l}} \geq 1-\frac{\varepsilon}{2}$$

is satisfied.

Claim 16 (Original): A unit according to claim 14, wherein the set of integral elements is a set of residues, which are calculated in correspondence with the user identification number and take a plurality of integers relatively prime to each other as moduli.

Claim 17 (Original): A unit according to claim 14, wherein the set of integral elements is a set of numbers of elements which are calculated in correspondence with the user identification number and belong to an equivalence class defined by a parallel transformation.

Claim 18 (Original): A unit according to claim 14, wherein the set of integral elements is a set of numbers of elements which are calculated in correspondence with the user identification number and belong to an equivalence class defined by a parallel transformation, and

where p_i (i=1, 2, ..., k') is one same positive integer p, a condition of

$$k' = \frac{c}{2}(k+l) \leq \frac{p^k - 1}{p - 1}$$

is further satisfied.

Claim 19 (Currently Amended): A unit for determining a user identification number of a colluder by detecting an embedded code, comprising:

an input port configured to receive from an external device predetermined content having the embedded code embedded therein;

a detector configured to detect said embedded code from said predetermined content;

a code extractor configured to extract the embedded code from said predetermined content, the embedded code ~~concatenating~~ is a concatenation of component codes respectively generated in correspondence with an inputted user identification number and also being such that among k' component codes capable of expressing all sets of integral elements that are calculated with respect to a predetermined number of user identification numbers, k combinations of the k' component codes can uniquely express the user identification numbers;

a code divider configured to perform a division into extracted component codes;

a component code decoder configured to decode each of the extracted component codes;

a colluder number calculator configured to calculate a user identification number of a colluder from a decoding result of each of the component codes; and

an output port for outputting said user identification number, wherein

the component code decoding means includes a block dividing section for dividing each of the component codes into blocks, a counting section for counting a number of bits of "1" in every one of the blocks, a first determining section for determining whether or not a count value obtained by the counting section exceeds a first threshold value, a second determining section for determining whether or not the count value is smaller than a second threshold value, a minimum position selecting section for selecting a minimum block determined as exceeding the first threshold value by the first determining section, and a maximum position selecting section for selecting a maximum block determined as being smaller than the second threshold value, thereby to output a selection results of the minimum and maximum position selecting sections, as a decoding result.

Claims 20-22 (Canceled).

Claim 23 (Previously Presented): A unit according to claim 5, wherein the colluder number calculator generates at least one user identification number candidate having a possibility to be the user identification number of the colluder, from the plurality of residue pairs, selects at least one user identification number having a lower possibility to be erroneously detected as the user identification number of the colluder, among the candidate, and decides the selected user identification number as the user identification number of the colluder.

Claim 24 (Previously Presented): A unit according to claim 5, wherein the colluder number calculator sequentially generates a plurality of user identification number candidates having a possibility to be the user identification number of the colluder, from the plurality of residue pairs, determines whether a possibility to be erroneously detected as the user identification number of the colluder is high or low, with respect to the candidates, and decides all of user identification numbers that have the possibility determined to be low, as user identification numbers of colluders.

Claim 25 (Previously Presented): A unit according to claim 23, wherein the colluder number calculator obtains a number of those residues among all of the residues pairs that satisfy a congruence to the residues taking the plurality of integers as modulus, with respect to all user identification numbers, and generates a user identification number which makes the number to be a predetermined threshold value or more, as a user identification number candidate of the colluder.

Claim 26 (Previously Presented): A unit according to claim 24, wherein the colluder number calculator obtains a number of those residues among all of the residues pairs that satisfy a congruence to the residues taking the plurality of integers as modulus, with respect to all user identification numbers, and generates a user identification number which makes the number to be a predetermined threshold value or more, as a user identification number candidate of the colluder.

Claim 27 (Previously Presented): A unit according to claim 5, wherein the colluder number calculator includes a storage device which stores a plurality of user identification numbers having a lower possibility at which the plurality of user identification numbers are erroneously detected as the user identification numbers of the colluder, and decides a user identification number which coincides with at least one user identification number candidate having a possibility to be the user identification number of the colluder, generated from the plurality of residue pairs, among the user identification numbers stored in the storage device.

Claims 28-32 (Canceled).